

# MIDI Protocol

Craig Sapp

# Bytes

MIDI protocol and MIDI files are based on bytes

# Bytes

- Bytes are 8-digit binary digits (on/off states)

**00000000 — 11111111**

bit = Binary digit

- Total number of configurations of the 8 digits

**$2^8 = 256$**

- Representing as an unsigned integer

**0 — 255**

- Representing as a 2's complement signed integer (still 256 states)

**0 — 127, -128 — -1**

# Positional notation of numbers

[https://en.wikipedia.org/wiki/Positional\\_notation](https://en.wikipedia.org/wiki/Positional_notation)

- What does “365” mean?

# Positional notation of numbers

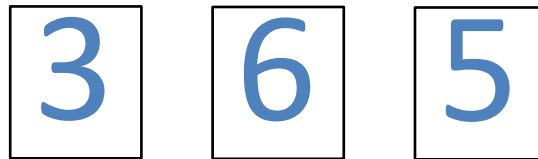
[https://en.wikipedia.org/wiki/Positional\\_notation](https://en.wikipedia.org/wiki/Positional_notation)

- What does “365” mean?

$$365 = 300 + 60 + 5$$

$$365 = 3 \times 100 + 6 \times 10 + 5 \times 1$$

$$365 = 3 \times 10^2 + 6 \times 10^1 + 5 \times 10^0$$



Digit position:

2      1      0



# Binary Numbers

What is the binary number 10110 in decimal positional notation (base-10)?

$$10110_2$$
$$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$16 + 0 + 4 + 2 + 0$$

$$22_{10}$$

# Hexadecimal Numbers

- Hexadecimal uses **16** as the positional base.
- Digits representing “10” through “15” are the letters “A” through “F”

$$A_{16} = 10_{10}$$

$$D_{16} = 13_{10}$$

$$B_{16} = 11_{10}$$

$$E_{16} = 14_{10}$$

$$C_{16} = 12_{10}$$

$$F_{16} = 15_{10}$$

# Hex vs Binary vs Decimal

- Computers operate internally with binary (on/off states)
- Converting  $01101010_2$  to decimal is non-trivial (add lots of powers of two)
- Converting  $01101010_2$  to hexadecimal is trivial (memorize 16 conversions) because every four binary digits represent one hex digit (“nibble”)

$01101010_2$  :

**$01101010_2$**   
**0110    1010**  
**4+2    8+2**  
**6        A**  
 **$6A_{16}$**

**$01101010_2$**   
**64+32+8+2**  
 **$106_{10}$**



# Useful Conversion to Know

$$11111111_2 = FF_{16} = 255_{10}$$

2's compliment interpretation:  $= -1_{10}$

$$01111111_2 = 7F_{16} = 127_{10}$$

$$10000000_2 = 80_{16} = 128_{10}$$

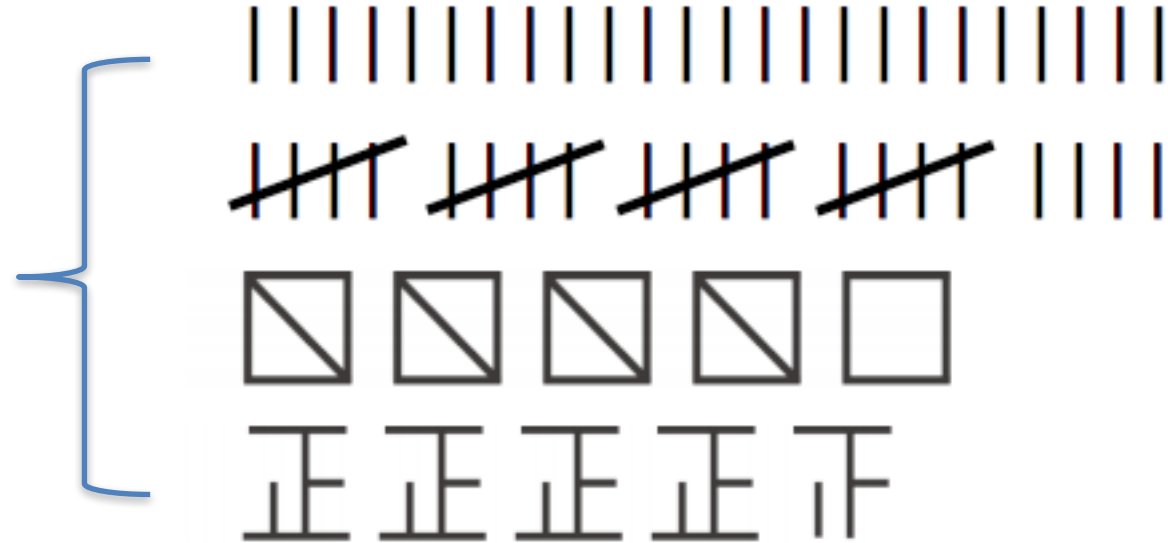
*Note:* FF<sub>h</sub> and 0xFF are alternate ways of indicating hex.

# Other Positional Bases

[http://wiki.ccarh.org/images/9/92/Hexadecimal\\_numbers.pdf](http://wiki.ccarh.org/images/9/92/Hexadecimal_numbers.pdf)

Tally marks (base-1):

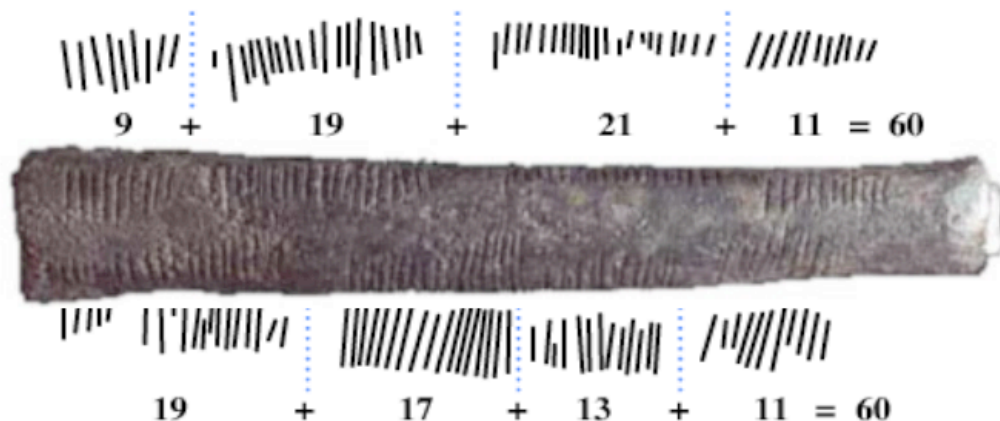
24



Ishango bone

~20,000 years old

[www.wikipedia.org/wiki/Ishango\\_bone](http://www.wikipedia.org/wiki/Ishango_bone)



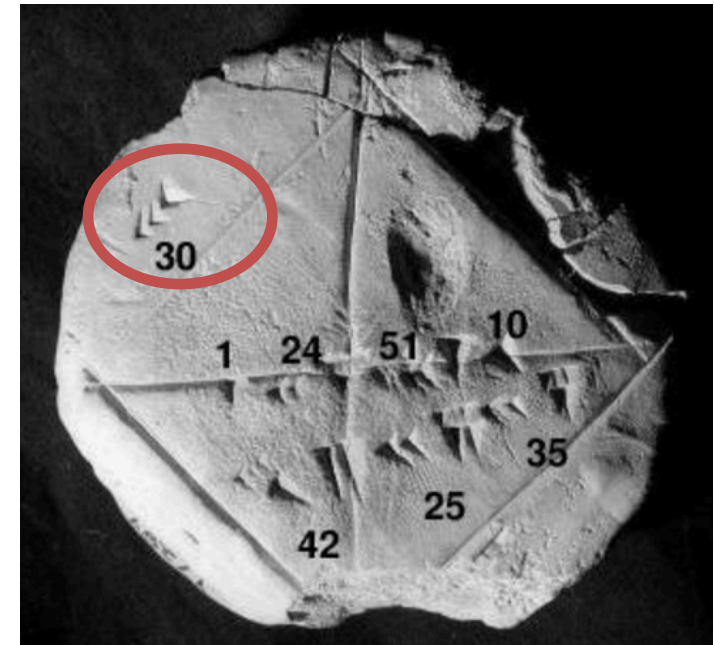
# Sexagesimal (Base-60)

<https://en.wikipedia.org/wiki/Sexagesimal>

Developed ~5000 years ago by Sumerians

(c. 1800–1600 BCE)

𐎶 1	𐎶𐎵 11	𐎶𐎵𐎶 21	𐎶𐎵𐎶𐎵 31	𐎶𐎵𐎶𐎵𐎶 41	𐎶𐎵𐎶𐎵𐎶𐎵 51
𐎶𐎶 2	𐎶𐎶𐎵 12	𐎶𐎶𐎶 22	𐎶𐎶𐎶𐎵 32	𐎶𐎶𐎶𐎵𐎶 42	𐎶𐎶𐎶𐎵𐎶𐎵 52
𐎶𐎶𐎶 3	𐎶𐎶𐎶𐎵 13	𐎶𐎶𐎶𐎶 23	𐎶𐎶𐎶𐎶𐎵 33	𐎶𐎶𐎶𐎶𐎵𐎶 43	𐎶𐎶𐎶𐎶𐎵𐎶𐎵 53
𐎶𐎶𐎶𐎶 4	𐎶𐎶𐎶𐎶𐎵 14	𐎶𐎶𐎶𐎶𐎶 24	𐎶𐎶𐎶𐎶𐎶𐎵 34	𐎶𐎶𐎶𐎶𐎶𐎵𐎶 44	𐎶𐎶𐎶𐎶𐎶𐎵𐎶𐎵 54
𐎶𐎶𐎶𐎶𐎶 5	𐎶𐎶𐎶𐎶𐎶𐎵 15	𐎶𐎶𐎶𐎶𐎶𐎶 25	𐎶𐎶𐎶𐎶𐎶𐎶𐎵 35	𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶 45	𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶𐎵 55
𐎶𐎶𐎶𐎶𐎶𐎶 6	𐎶𐎶𐎶𐎶𐎶𐎶𐎵 16	𐎶𐎶𐎶𐎶𐎶𐎶𐎶 26	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 36	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶 46	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶𐎵 56
𐎶𐎶𐎶𐎶𐎶𐎶𐎶 7	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 17	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 27	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 37	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶 47	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶𐎵 57
𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 8	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 18	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 28	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 38	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶 48	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶𐎵 58
𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 9	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 19	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 29	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 39	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶 49	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶𐎵 59
𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 10	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 20	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶 30	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵 40	𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎶𐎵𐎶 50	



$$1;24,51,10 = 1 + 24/60^1 + 51/60^2 + 10/60^3 \\ = 1.41421296 \approx \sqrt{2}$$

Minutes/Seconds:

$$12'30'' + 16'42'' = 29'12''$$

# Mayan Numbers (Base-20)



0



1



2



3



4



5



6



7



8



9



10



11



12



13



14



15



16



17



18



19

# Mayan Numbers (2)



$$3 \times 20^2 = 1200$$



$$0 \times 20^1 = 0$$



$$18 \times 20^0 = 18$$

1218

# MIDI Bytes

# MIDI Data/Command Bytes

0 — 127

128 — 255

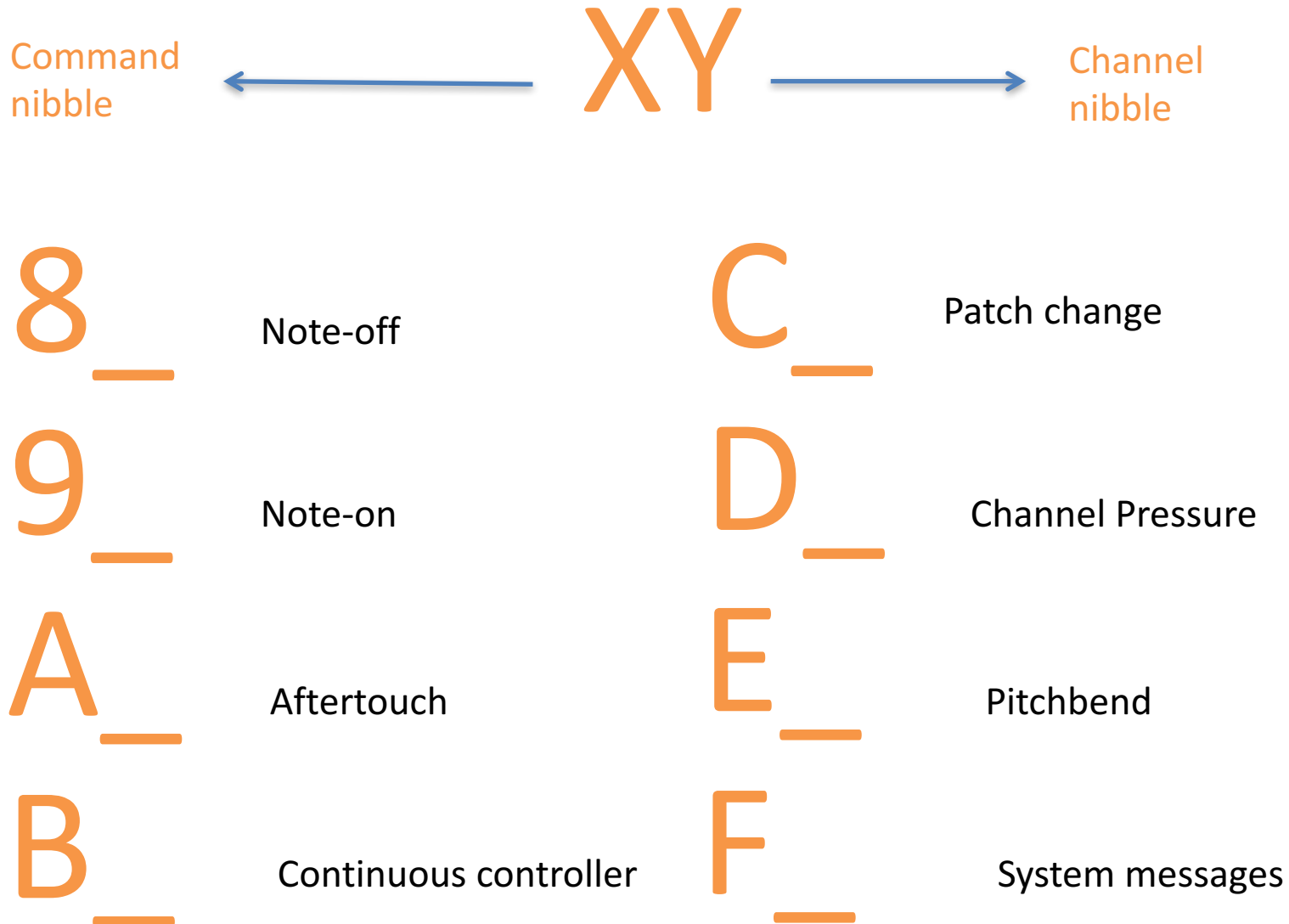
Data bytes

Command bytes

$00000000_2$  —  $01111111_2$        $10000000_2$  —  $01111111_2$   
00h — 7Fh      80h — FFh

7-BITS

# MIDI Commands





# MIDI Command Parameters

 = data byte (number between 0 and 127)

Note-off	8	key	velocity	Patch change	C	instrument	
Note-on	9	key	velocity	Channel pressure	D	pressure	
Aftertouch	A	key	pressure	Pitchbend	E	LSB	MSB
Continuous controller	B	controller	value	System messages	F	( u s u a	

# Running Status

REGULAR MESSAGING:


90 3C 48 91 3E 52 91 3E 00 90 3C 00

RUNNING STATUS (remove repeated command bytes):

90 3C 48 91 3E 52 3E 00 90 3C 00

Reconstructing regular messages

90 3C 48 91 3E 52 (91) 3E 00 90 3C 00



# Alternate Note-Off Commands

8\_ commands are for note-off messages:

80 3C 64

= turn off note 60 (0x3C, middle C)  
With a release velocity of 100 (64h)

But also a common note-off shorthand:

90 3C 00

= turn off note 60 (0x3C, middle C)  
With undefined release velocity

- So softest sounding note has data byte of 01, not 00.

# Cinmidi

<http://wiki.ccarh.org/wiki/Cinmidi>

“Console-In MIDI”: Display incoming MIDI messages in terminal with timestamps.

```
;;
;; Style:                default
;; Timing:               delta milliseconds
;; Message format:      delta-time, MIDI command-byte, MIDI parameter-byte(s)
;; Format:               asciimidi 1.0
;; Command-line:        cinmidi -o invention13-28.txt -p 1
;; Input Port:          1:
;; Cpu Speed:           1000 MHz
;;
```

```
0          0x90  64  88          ; NOTE chan:1 key:E4 vel:88
30         0x80  45  64          ; NOTEOFF chan: 1 key:A2 vel: 64
128        0x90  69  88          ; NOTE chan:1 key:A4 vel:88
7          0x80  64  64          ; NOTEOFF chan: 1 key:E4 vel: 64
23         0x90  57  87          ; NOTE chan:1 key:A3 vel:87
109        0x90  72  91          ; NOTE chan:1 key:C5 vel:91
25         0x80  69  64          ; NOTEOFF chan: 1 key:A4 vel: 64
98         0x90  71  91          ; NOTE chan:1 key:B4 vel:91
4          0x80  72  64          ; NOTEOFF chan: 1 key:C5 vel: 64
102        0x80  71  64          ; NOTEOFF chan: 1 key:B4 vel: 64
28         0x90  64 104          ; NOTE chan:1 key:E4 vel:104
109        0x80  57  64          ; NOTEOFF chan: 1 key:A3 vel: 64
```